

In transit data processing for extreme scale computing

Scott A. Klasky

+



H. Abbasi², Q. Liu², J. Logan¹, M. Parashar⁶, N. Podhorszki¹, K. Schwan⁴, A. Shoshani³, M. Wolf⁴, S. Ahern¹, I. Altintas⁹, W. Bethel³, L. Chacon¹, CS Chang¹⁰, J. Chen⁵, H. Childs³, J. Cummings¹³, C. Docan⁶, G. Eisenhauer⁴, S. Ethier¹⁰, R. Grout⁷, R. Kendall¹, J. Kim³, T. Kordenbrock⁵, Z. Lin¹⁰, J. Lofstead⁵, X. Ma¹⁵, K. Moreland⁵, R. Oldfield⁵, V. Pascucci¹², N. Samatova¹⁵, W. Schroeder⁸, R. Tchoua¹, Y. Tian¹⁴, R. Vatsavai¹, M. Vouk¹⁵, K. Wu³, W. Yu¹⁴, F. Zhang⁶, F. Zheng⁴ + many many many more!

¹ORNL, ²U.T. Knoxville, ³LBL, ⁴Georgia Tech, ⁵Sandia Labs, ⁶Rutgers, ⁷NREL, ⁸Kitware, ⁹UCSD, ¹⁰PPPL, ¹¹UC Irvine, ¹²U. Utah, ¹³Caltech, ¹⁴Auburn University, ¹⁵NCSU

Main points I want to make

- Creating a computational laboratory is really difficult
 - Problems are really hard, Multi-scale, Multi-physics
- Complexities come from
 - Systems
 - Applications
 - Rapidly changing requirements
 - Evolving target platforms
 - Diverse Team
- And we never heard this before....

It's all new this time around?



National Science Foundation
WHERE DISCOVERIES BEGIN

NSF Web Site



Home Funding Awards Discoveries **News** Publications Statistics About FastLane

News



News

News From the Field

For the News Media

Special Reports

Research Overviews

NSF-Wide Investments

Speeches & Lectures

NSF Current Newsletter

Multimedia Gallery

News Archive

- Press Releases
- Media Advisories
- News Tips
- Press Statements
- Speech Archives
- **Frontiers Archives**

Frontiers

Crash Testing Black Holes: A Colossal Collision

July/August 1996

What do you get when two black holes crash into each other?

A bigger black hole!

At least that's what University of Texas physics collaborators got when they built a computer

Theoretically, a black hole is a collapsed star even light can't escape. While the idea is wide evidence of the phenomenon. This and other Grand Challenge Binary Black Hole Alliance, a black hole collision when they find them.

The simulation starts with two symmetrical stars. When the program ends, only one black hole survives the battle for supremacy.

By definition, a black hole has an irresistible escape from its symmetrical boundary. But an equally strong gravitational field.

Each hole pulls on the other. Like the moon's yanks its counterpart out of the symmetrical tear-shaped, and then oblong -- stretching and

Finally the holes' tips touch and they collapse into one hole.

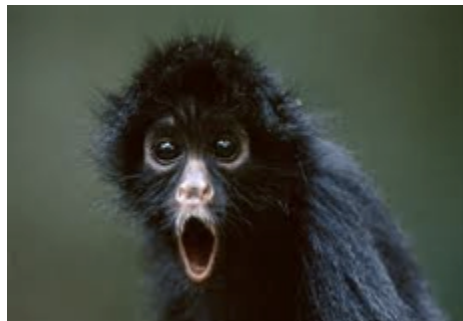
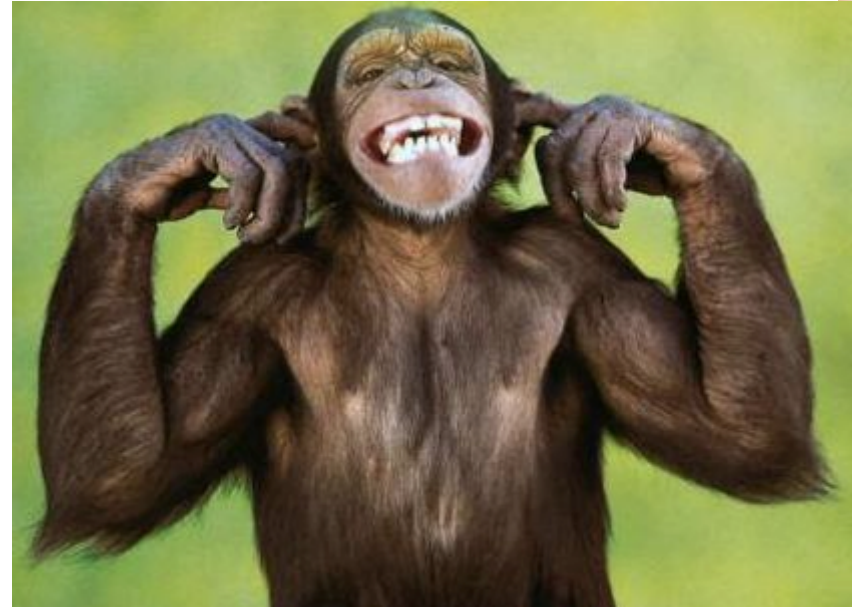
What is so difficult?

- The problem has at least multiple scales
 - natural scale near each hole $\sim m$
 - ringdown natural scale $\sim 20 (m_1 + m_2)$
 - inspiral wavelength $\sim \text{hundreds} \times (m_1 + m_2)$

Need Approach to Handle Different Scales ! ! ! !

Some people think I/O is only about reads/writes

- But why do we want to read/write?
- Is checkpoint-restart it?
- Some people say that you can re-run your code, and place analytics inside the code.
 - \$150K/day to run a code, what will the cost be for the exascale?
- Some people say that they can just program I/O + analytics on their own.



Some people think we just need to collaborate

S3D



+

Boxlib



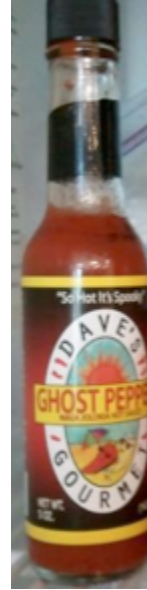
+

ROSE



+

ADIOS



+

DSL



=



Or



My beliefs for exascale

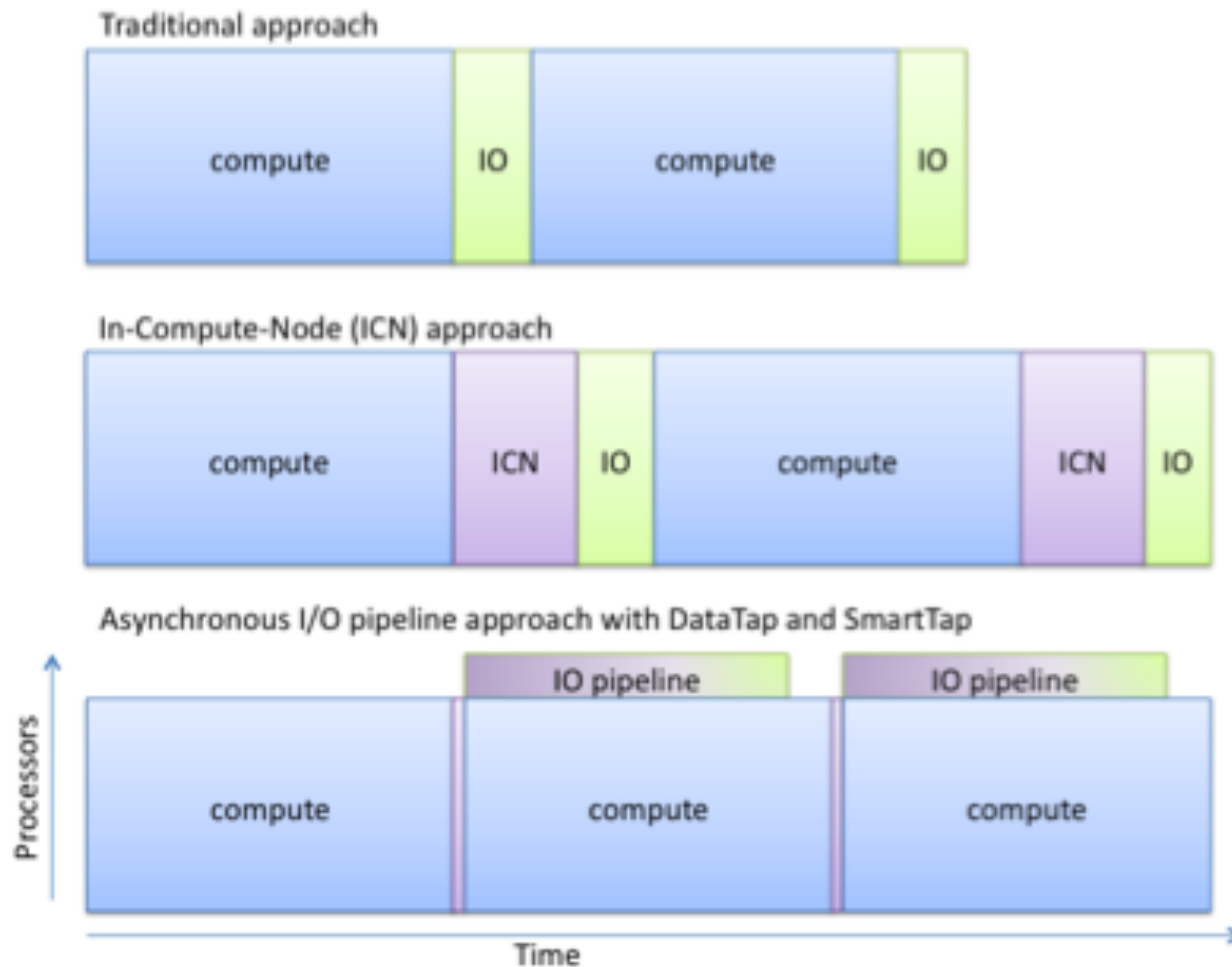
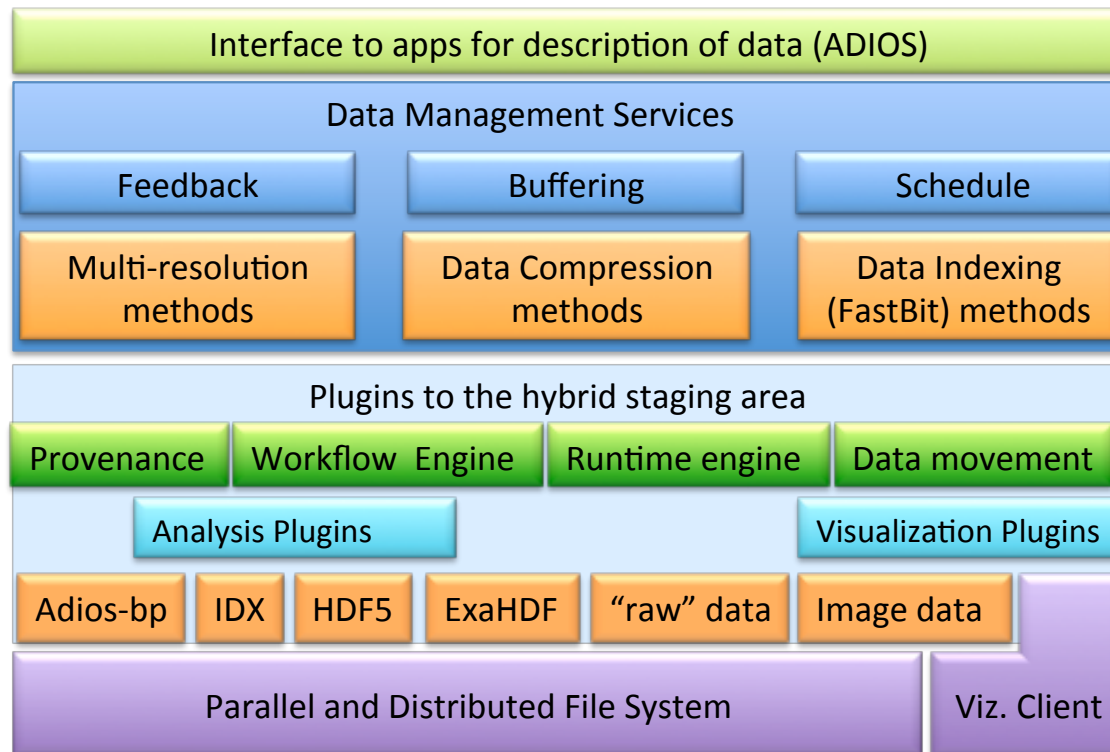


Fig. 5. Different placements of I/O activities in time and processor space

Use principles from the Enterprise

- SOA --- Yes that means Service Oriented Architecture
- But.... For HPC/ I mean Exascale.
- Our approach:

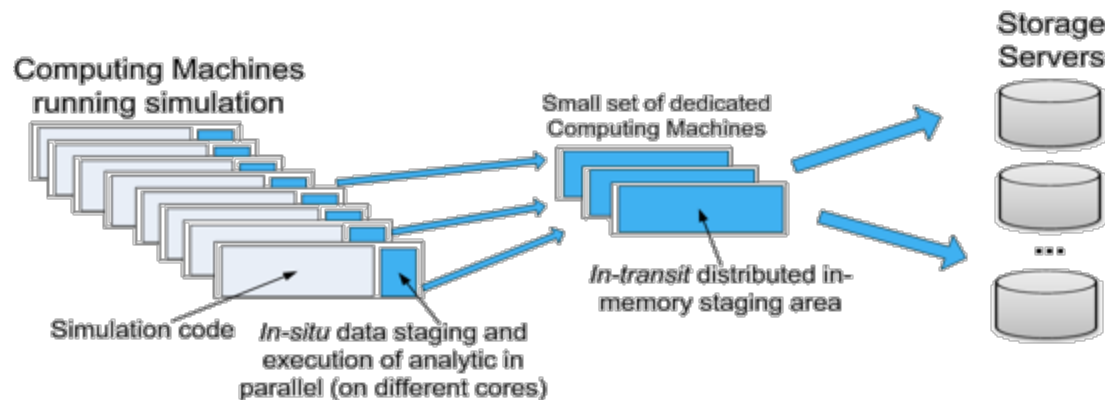


But keep an open mind to new ideas



My belief for exascale (from Exact)

- ADIOS is being implemented as a Service Oriented Architecture
 - Create an environment where applications can abstract data movement/IO as read/writes/queries



- Exploit complex-memory hierarchies with **Hybrid Data Staging** to:
 - Decrease the gap between CPU and IO speeds
 - Dynamically deploy and execute data analytical or pre-processing operations either in-situ or in-transit
 - Improved IO write performance

Co-design the analytics pipeline

- **Optimization:**
- **Where/How should analytics be run?**
 - Inline with simulation?
 - Separate cores? Helper cores?
 - Separate staging nodes?
 - Offline?
- **Guided by what metrics?**
 - *Performance:* **Total Execution Time** of both simulation and analytics
 - *Cost:* **CPU hours** charged for simulation and analytics
 - *Time to Data:* **Delay** between data generation and analytics results
 - *Data movement* ~ **Power** consumption

